四川師範大學
SICHUAN NORMAL UNIVERSITY

第5章　数组和广义表

DATA STRUCTURE

计算机科学学院　廖雪花

本章内容简介

数组和广义表

- 5.1 数组的定义
- 5.2 数组的顺序表示和实现
- 5.3 矩阵的压缩存储
- 5.4 广义表的定义
- 5.5 广义表的存储结构

5.3 矩阵的压缩存储

廖雪花　LiaoXuehua

四川師範大學
SICHUAN NORMAL UNIVERSITY

矩阵的压缩存储

- 两类矩阵的压缩存储
  - 特殊矩阵
  - 稀疏矩阵

5.3.2 稀疏矩阵的压缩存储

- 什么是稀疏矩阵?
  - 在上节提到的特殊矩阵中，元素的分布呈现某种规律，故一定能找到一种合适的方法，将它们进行压缩存放。
  - 但是，在实际应用中，我们还经常会遇到一类矩阵：其矩阵阶数很大，非零元个数较少，零元很多，但非零元的排列没有一定规律，我们称这一类矩阵为稀疏矩阵。
  - 一般地：
    - 稀疏因子$\delta = t/(m\times n) \leq 0.05$的矩阵称为稀疏矩阵。

5.3.2 稀疏矩阵的压缩存储

- 问题：
  - 以常规方法，即以二维数组表示高阶的稀疏矩阵时产生的问题。
    - 零值元素占了很大空间；
    - 计算中进行了很多和零值的运算，遇除法，还需判别除数是否为零。

5.3.2 稀疏矩阵的压缩存储

- 解决问题的原则：
  - 尽可能少存或不存零值元素。
  - 尽可能减少没有实际意义的运算。
  - 操作方便。 即：
    - 能尽可能快地找到与下标值(i，j)对应的元素；
    - 能尽可能快地找到同一行或同一列的非零值元。

5.3.2 稀疏矩阵

抽象数据类型稀疏矩阵的定义

ADT SparseMatrix{

数据对象：$D = \{a_{ij} | a_{ij} \in ElemSet, 1 \le i \le m, 1 \le j \le n\}$

数据关系：$S = \{Row, Col\}$

$Row = \{< a_{i,j}, a_{i,j+1} > | 1 \le i \le m, 1 \le j < n\}$

$Col = \{< a_{i,j}, a_{i+1,j} > | 1 \le i < m, 1 \le j \le n\}$

基本操作：

CreateSMatrix(&M)	DestroySMatrix(&M)
PrintSMatrix(M)	CopySMatrix(M,&T)
AddSMatrix(M,N,&Q)	SubSMatrix(M,N,&Q)
MultSMatrix(M,N,&Q)	TransposeSMatrix(M,&T)

} ADT SparseMatrix

5.3.2 稀疏矩阵

- 稀疏矩阵的压缩存储
  - 基本思想
    - 按照压缩存储的概念，要存放稀疏矩阵的元素，由于没有某种规律，除存放非零元的值外，还必须存贮适当的辅助信息，才能迅速确定一个非零元是矩阵中的哪一个位置上的元素。
  - 几种压缩存储方案
    - 三元组顺序表
    - 带行逻辑链接的顺序表
    - 十字链表

一、三元组顺序表存储

三元组顺序表

- 三元组
  - 非零元的行号i、列号j、值e构成一个三元组（i,j,e）。
- 三元组表示法
  - 整个稀疏矩阵中非零元的三元组合起来称为三元组表。
  - 一般规定以行序为主序，采用顺序存储结构依次存储稀疏矩阵中所有的非0元的三元组（i,j,e），构成叙述矩阵的三元组顺序表。

例1：稀疏矩阵的压缩存储

$$\begin{vmatrix} 0 & 12 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 24 & 0 & 0 & 0 & 0 \\ 0 & 18 & 0 & 0 & 0 & 0 & 0 \\ 15 & 0 & 0 & -7 & 0 & 0 & 0 \end{vmatrix}$$

i	*j*	*e*
1	2	12
1	3	9
3	1	-3
3	6	14
4	3	24
5	2	18
6	1	15
6	4	-7

mu=6 *nu*=7 *tu*=8

稀疏矩阵的三元组顺序表存储表示

```
#define  MAXSIZE  <非0元素个数的最大值>
typedef struct{
        int i,j;
        ElemType e;
}Triple;
typedef struct{
        Triple data[MAXSIZE+1];
        int mu,nu,tu;
}TSMatrix;
```

例2：稀疏矩阵的三元组顺序表

$$M=\begin{vmatrix} 0 & 12 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 24 & 0 & 0 & 0 & 0 \\ 0 & 18 & 0 & 0 & 0 & 0 & 0 \\ 15 & 0 & 0 & -7 & 0 & 0 & 0 \end{vmatrix}$$

	i	*j*	*e*
M.data[1]	1	2	12
	1	3	9
	3	1	-3
	3	6	14
	4	3	24
	5	2	18
	6	1	15
	6	4	-7

M.mu＝6，M.nu＝7， M.tu＝8

稀疏矩阵的转置运算

- 例如：

$$M=\begin{vmatrix} 0 & 12 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 24 & 0 & 0 & 0 & 0 \\ 0 & 18 & 0 & 0 & 0 & 0 & 0 \\ 15 & 0 & 0 & -7 & 0 & 0 & 0 \end{vmatrix} \qquad T=\begin{vmatrix} 0 & 0 & -3 & 0 & 0 & 15 \\ 12 & 0 & 0 & 0 & 18 & 0 \\ 9 & 0 & 0 & 24 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -7 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 14 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

对应的稀疏矩阵如下：

M.data[1]

i	j	e
1	2	12
1	3	9
3	1	-3
3	6	14
4	3	24
5	2	18
6	1	15
6	4	-7

M.mu＝6，M.nu＝7， M.tu＝8

T.data[1]

i	j	e
1	3	-3
1	6	15
2	1	12
2	5	18
3	1	9
3	4	24
4	6	-7
6	3	14

T.mu＝7，T.nu＝6， T.tu＝8

稀疏矩阵的转置运算

- 问题：
  - 在三元组表表示的稀疏矩阵中，怎样求得它的转置呢?
- 分析
  - 方法1：按*M*的列序进行转置
  - 方法2：按*M*的行序进行转置（快速转置）

稀疏矩阵的转置运算

- 方法1：按照M的列序进行转置
  - 分析：
    - 由于M的列即为T的行，在M.data中，按列扫描，则得到的T.data必按行优先存放。
    - 但为了找到M的每一列中所有的非零的元素，每次都必须从头到尾扫描M的三元组表(有多少列，则扫描多少遍)。
  - 算法描述如下（算法5.1）：

稀疏矩阵的转置运算

```
Status TransposeSMatrix(TSMatrix M,TSMatrxi &T){
 T.mu=M.nu;T.nu=M.mu;T.tu=M.tu;
 if (T.tu){
    q=1;
    for (col=1;col<=M.nu;col++)
      for (p=1;p<=M.tu;p++)
          if (M.data[p].i==col){
                T.data[q].i=M.data[p].j;
                T.data[q].j=M.data[p].i;
                T.data[q].e=M.data[p].e;
                ++q;
              }
 }
 return OK;
}// TransposeSMatrix
```

时间复杂度：O(nu×tu)

适用于tu＜＜mu×nu的情况。

稀疏矩阵的转置运算

- 方法2：按照M的行序进行转置（快速转置）
  - 分析：
    - 即按M.data中三元组的次序进行转置，并将转置后的三元组放入T中恰当的位置。按照M中三元组的顺序进行转置，并将转置后的三元组置入T中恰当的位置。
    - 引入两个辅助向量：
      - num[col]:表M中第col列中非0元的个数；
      - cpot[col]:表M中第col列的第一个非0元在T.data中的位置。

稀疏矩阵的转置运算

- 例如：

$$M = \begin{vmatrix} 0 & 12 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 24 & 0 & 0 & 0 & 0 \\ 0 & 18 & 0 & 0 & 0 & 0 & 0 \\ 15 & 0 & 0 & -7 & 0 & 0 & 0 \end{vmatrix}$$

col	1	2	3	4	5	6	7
num[col]	2	2	2	1	0	1	0
cpot[col]	1	3	5	7	8	8	9

一般地：

$$\begin{cases} cpot[1] = 1 \\ cpot[col] = cpot[col-1] + num[col-1] \quad 2 \le col \le m.nu \end{cases}$$

稀疏矩阵的快速转置运算

```
Status FastTransposeSMatrix( TSMatix M,TSMatrix &T){
    T.mu=M.nu;T.nu=M.mu;T.tu=M.tu;
    if (T.tu){
        for (col=1;col<=M.nu;col++) num[col]=0;
        for (t=1;t<=M.tu;t++)   ++num[M.data[t].j];
        cpot[1]=1;
        for (col=2;col<=M.nu;col++)
            copt[col]=cpot[col-1]+num[col-1];
```

稀疏矩阵的快速转置运算

```
        for (p=1;p<=M.tu;p++) {
                col=M.data[p].j;  q=cpot[col];
                T.data[q].i=M.data[p].j;
                T.data[q].j=M.data[p].i;
                T.data[q].e=M.data[p].e;
                ++cpot[col];
        }//for
    }//if
     return OK;
} //FastTransposeSMatrix
```

时间复杂度O（ nu＋tu)

本节要点

- 稀疏矩阵的压缩存储：
  - ✓ 稀疏矩阵的定
  - ✓ 稀疏矩阵的存储：三元组顺序表、带行逻辑链接的顺序表、十字链表
- 三元组顺序表：
  - ✓ 稀疏矩阵的三元组顺序表存储表示
  - ✓ 按列序矩阵转置
  - ✓ 按行序矩阵转置（快速转置）

四川師範大學
SICHUAN NORMAL UNIVERSITY

感谢聆听